

(19)



(11)

EP 3 405 873 B1

(12)

EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention of the grant of the patent:
07.12.2022 Bulletin 2022/49

(51) International Patent Classification (IPC):
G06F 12/02 ^(2006.01) **G06F 11/10** ^(2006.01)
G11C 11/00 ^(2006.01) **G11C 29/00** ^(2006.01)

(21) Application number: **17741826.6**

(52) Cooperative Patent Classification (CPC):
G06F 11/1068; G06F 12/0246; G11C 5/04;
G11C 5/141; G11C 7/10; G11C 7/22; G11C 11/005;
G11C 16/32; G11C 29/883; G06F 2212/1032;
G06F 2212/7202; G11C 2029/0409;
G11C 2029/0411; G11C 2207/2245; Y02D 10/00

(22) Date of filing: **18.01.2017**

(86) International application number:
PCT/US2017/013859

(87) International publication number:
WO 2017/127386 (27.07.2017 Gazette 2017/30)

(54) **NON-VOLATILE MEMORY MODULE ARCHITECTURE TO SUPPORT MEMORY ERROR CORRECTION**

NICHTFLÜCHTIGE SPEICHERMODULARCHITEKTUR ZUR UNTERSTÜTZUNG DER SPEICHERFEHLERKORREKTUR

ARCHITECTURE DE MODULE DE MÉMOIRE NON VOLATILE POUR PRENDRE EN CHARGE UNE CORRECTION D'ERREURS DE MÉMOIRE

(84) Designated Contracting States:
AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR

- **PARRY, Jonathan**
Boise
Idaho 83709 (US)

(30) Priority: **19.01.2016 US 201615000812**

(74) Representative: **Marks & Clerk LLP**
15 Fetter Lane
London EC4A 1BW (GB)

(43) Date of publication of application:
28.11.2018 Bulletin 2018/48

(56) References cited:
US-A1- 2002 080 541 US-A1- 2008 126 690
US-A1- 2009 292 860 US-A1- 2010 205 470
US-A1- 2011 239 021 US-A1- 2012 297 140
US-A1- 2013 138 874 US-A1- 2014 215 277
US-A1- 2014 244 900 US-A1- 2014 244 900
US-A1- 2014 258 601 US-B2- 7 162 569
US-B2- 7 162 569

(73) Proprietor: **Micron Technology, Inc.**
Boise, ID 83716-9632 (US)

(72) Inventors:

- **PAX, George**
Boise
Idaho 83716 (US)

EP 3 405 873 B1

Note: Within nine months of the publication of the mention of the grant of the European patent in the European Patent Bulletin, any person may give notice to the European Patent Office of opposition to that patent, in accordance with the Implementing Regulations. Notice of opposition shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

DescriptionPriority Application

[0001] This applications claims the benefit of priority to U.S. Application Serial No. 15/000,812, filed 19 January 2016.

Background

[0002] As memory devices of all types have evolved, continuous strides have been made in improving their performance in a variety of respects. For example, the storage capacity of memory devices has continued to increase. This increased capacity, has allowed the development of ever more sophisticated applications and data structures. Many enterprises rely on computer systems that run around the clock to hold and have available critical data for the continued operation of the enterprise. Although highly robust, memory chips can fail. Without a backup plan, data stored on a failed memory chip can be of such critical importance to the operation of a computer system that significant system downtime can result if the information cannot be reassembled in a timely manner. Current memory devices can include volatile memory for normal operation and non-volatile memory for situations where information stored in volatile memory needs to be backed-up, such as when power to the volatile memory can be interrupted. Under such situations, data can be transferred from the volatile memory to the non-volatile memory for storage during the power disruption. Upon subsequent power restoration the data can be retrieved back to the volatile memory for fast storage and retrieval with a host system. Upon failure of a non-volatile memory component of the memory module, data associated with the failed memory component can be lost and the computer system can be rendered inoperable. Document US 2005/0146939 discusses that data is transferred from a controller to a first memory chip and a programming operation is caused to begin. While that first memory chip is busy performing that program operation, data is transferred from the controller to a second memory chip and a programming operation is caused to begin in that chip. Data transfer can begin to the first memory chip again once it has completed its programming operation even though the second chip is still busy performing its program operation.

Summary

[0003] The invention is defined by the method of claim 1 and the system of claim 6. Advantageous embodiments are defined in the dependent claims.

Brief Description of the Drawings

[0004] In the drawings, which are not necessarily drawn to scale, like numerals may describe similar com-

ponents in different views. Like numerals having different letter suffixes may represent different instances of similar components. The drawings illustrate generally, by way of example, but not by way of limitation, various embodiments discussed in the present document.

FIG. 1 illustrates generally an example memory module.

FIG. 2 illustrates generally an example data lane to non-volatile memory page mapping for a memory device according to various examples of the present subject matter.

FIG. 3 illustrates generally an alternative example memory module.

FIG. 4 illustrates generally a flowchart of an example method for transferring data between a volatile memory system and a non-volatile memory system.

FIG. 5 illustrates generally an example data lane to non-volatile memory page mapping for a memory module according to various examples of the present subject matter.

FIG. 6 illustrates generally a flowchart of an example method for transferring data between a volatile memory system and a non-volatile memory system.

Description of the Embodiments

[0005] The present inventors have recognized apparatus and methods for reliably storing and retrieving data within a non-volatile memory component of a memory module. In addition, stored data can be reconstructed using error correcting code (ECC) information such that if a non-volatile memory chip of the memory module fails, the data, or information, stored on the failed chip can be restored and the memory module or host system can mark the chip as failed and cease using the failed memory chip, whether volatile or non-volatile. Currently, non-volatile dual in-line memory modules (NVDIMM) transfer data to and from the non-volatile memory devices in a way that may not support memory error correction such as, but not limited to, ECC support, chip kill functionality or combinations thereof. The present subject matter, in addition to discussing the physical architecture of example NVDIMM also discusses data transfer schemes within memory modules that better support chip kill functionality. Chip kill functionality, in general, includes a chip kill algorithm executed by the host system that can use ECC information and the other valid data in the memory devices to reconstruct or restore data corrupted by a memory chip failure. Network server systems have included chip kill functionality such that a failed volatile memory chip can be identified by the host processor, the failed chip can be marked not to use, the corrupted data can be restored and remapped, and the system can continue to function until the failed non-volatile memory chip can be replaced. The present subject matter provides memory modules and methods that can extend chip kill functionality such that the host system can identify failed non-

volatile memory devices, restore corrupted data resulting from the non-volatile memory device failure, and in certain examples, continue to operate the system until the non-volatile memory device can be replaced.

[0006] FIG. 1 illustrates generally an example memory module 100. In certain examples, the memory module 100 can include, a power supply 101, at least one non-volatile memory 102, a non-volatile memory controller 103, a register 104, a plurality of volatile memory devices 105 (e.g., dynamic random access memory (DRAM) devices), a control and address multiplexer 106, a plurality of data multiplexers 107, and a plurality of interconnects (APR, BPR, MDQ). In certain examples, the memory can be formed on a single printed-circuit board (PCB). In certain examples, when system power from a system power supply (not shown) is removed from the memory module 100, the memory device power supply 101 can provide sufficient power to allow data to be moved from the volatile memory devices 105 to the non-volatile memory 102. In certain examples, the power supply 101 or backup power source can include a battery. In some examples, the power supply 101 can include a capacitor. In certain examples, the memory module 100 is representative of either a DDR3 or a DDR4 memory module.

[0007] The non-volatile memory may be termed a "bank" having one or more individual non-volatile memory devices. For simplicity, the description herein will refer to a "non-volatile memory or non-volatile memory device," with the understanding that in all cases the referenced non-volatile memory may be a bank of one or more individual non-volatile memory devices. The non-volatile memory 102 can save and maintain data when a power source external to the memory module 100 is removed. Examples of non-volatile memory devices can include, but are not limited to, charge storage type memories including various flash memory technologies such as NAND flash memories, magnetic random access memory (MRAM), spin-transfer torque random-access memory (STT-RAM), ferroelectric random-access memory (FeRAM), phase-change memory (PCM), resistive random-access memory (RRAM), or combinations thereof. In certain examples, the volatile memory devices 105 can include random access memory including, but not limited to, dynamic random access memory (DRAM) that can provide relatively faster read and write speeds for use by a host system (e.g., host processor). Such volatile random access memory typically cannot maintain stored data when system power is removed from the memory module 100. The register 104 can control access to the volatile memory devices 105.

[0008] When system power is present, the register 104 can receive command and address control information such as memory requests, for example, from the host system via a host command and address bus 108, and the command and address multiplexer 106. In certain examples, the host command and address bus 108 can be coupled to external connectors for control formed on the printed-circuit board. The register 104 can respond

to command and address control information by controlling the volatile memory devices 105 to present data to the host or receive data from the host, via the data multiplexers 107 and an external connector, for storage in the volatile memory devices 105. In certain examples, error correction code (ECC) information can be routed by the register 104 for storage in the volatile memory devices 105.

[0009] In certain examples, the memory module 100 can include two volatile memory command and address buses (APR, BPR), each bus coupled to a respective block 109, 110 of multiple volatile memory devices 105 and configured to transfer command and address control information to the volatile memory devices 105. In some examples, a first block 109 of volatile memory devices 105 can include five data lanes and a second block 110 of volatile memory devices 105 can include four data lanes. In certain examples, the register 104 can provide command and address control information to the volatile memory devices 105 to retrieve data requested by the host system or hold data provided by the host system. As used herein, a data lane is defined as a data channel that includes a data communication path and associated volatile memory associated with transferring data to and from a non-volatile memory controller. The number of data lanes in a memory module can be influenced by the number of volatile memory devices in the memory module, the number of memory modules that can simultaneously connect with the host data bus or connector, the size of the host data bus or connector interfacing with the memory module, the bit length of a data word associated with the volatile memory devices, or combinations thereof. In the examples of FIGS. 1 and 3, a data lane or data channel includes the memory and data path associated with a plurality of volatile memory devices to communicate with a host data bus. In the depicted example configurations, the data path further includes either a plurality of data multiplexers (Fig. 1), or a plurality of data buffers (Fig. 3) that can simultaneously communicate with a host data bus to facilitate the described communication therewith. In certain examples, a byte lane can be associated with one or more volatile memory devices (DRAM X, DRAM X+9) that can communicate with a host data bus via a corresponding data multiplexer (FIG. 1, MUX X) or a data buffer (FIG. 3, DBX). In certain examples, each data lane can be a 4-bit wide data lane (nibble lane). In some examples, each data lane can be 8-bit wide data lane (byte lane). It is understood that other data lane bit widths are possible without departing from the scope of the present subject matter.

[0010] In certain examples, the non-volatile memory controller 103 can control movement of data between the volatile memory devices 105 and the non-volatile memory device 102. For example, upon system power loss, the non-volatile memory controller 103 can switch the command and address multiplexer 106 to allow the register 104 to receive command and address control information from the non-volatile memory controller 103 via

a second command and address bus 111 instead of from the host system or host processor. The register 104 can then assist in controlling movement of data from the volatile memory devices 105 to a data bus (MDQ) that can couple the volatile memory devices 105 with the non-volatile memory controller 103. Data can be moved from the volatile memory devices 105 to the non-volatile memory controller 103 using the data bus (MDQ). In certain examples, the non-volatile memory controller 103 can provide routing information to the plurality of data multiplexers 107 so that the data from the volatile memory devices 105 is routed to the data bus (MDQ) and not to the host system or processor. In certain examples, the register 104 and the non-volatile memory controller 103 can cooperate to organize data storage in at least the non-volatile memory device 102 to allow for chip kill functionality. As data is received at the non-volatile memory controller, the non-volatile memory controller moves the data to the non-volatile memory device along with signaling to initiate programming of the non-volatile memory of the non-volatile memory device.

[0011] In certain examples, the memory module 100 can include an additional non-volatile memory devices (not shown) such as an EPROM that includes parameters that describe the memory device. In certain systems, the host can read the parameters stored on the EPROM and, upon recognizing that the memory module 100 has a NVDIMM architecture, can monitor and provide certain signals specific to the memory module 100. For example, the host may monitor the system power supply and provide a signal to the memory module 100 when an abnormality is detected. The memory module 100 can optionally use a state of the signal to initiate saving the data on the volatile memory devices 105 to the non-volatile memory device 102. In certain examples, a second state of the signal can initiate restoring the volatile memory data from the non-volatile memory.

[0012] In systems that use memory modules, such as those illustrated in FIGS. 1 and 3, the host processor, or processors, can store or hold data, including ECC information, in volatile memory. A system with chip kill functionality can allow reconstructing data of a failed volatile memory device 105 if the lost data does not represent a significant amount of sequential data. Existing methods of using a memory module can transfer data stored in volatile memory to the non-volatile memory. Such methods can include reading sequential data and ECC information simultaneously from each volatile memory data lane and storing the data and ECC information in the non-volatile memory device. More specifically, the non-volatile controller moves data in parallel from all the volatile memory data lanes to the non-volatile controller and then to a page of the non-volatile memory device. As the information is received at the non-volatile memory device, the information is sequentially programmed on a page of the non-volatile memory. If a page of non-volatile memory is lost due to a non-volatile memory chip failure, a large

chunk of sequential data and ECC information is lost. Because the data and ECC information represents a large chunk of sequential information, the information is not recoverable even in a system that supports chip kill.

[0013] As discussed above, chip kill functionality can allow recovery of data if there is a limited number of bit failures within a chunk of consecutive data. A system with chip kill functionality can allow reconstructing data of a failed memory chip, such as a failed volatile memory device 105, if the lost data does not represent a significant amount of sequential data. In certain examples of the present subject matter, the chip kill functionality that a system can have to identify and correct volatile memory failures can be extended to identify and correct data corrupted due to failures in the non-volatile memory device 102. Typical failures of a non-volatile memory device include page failures. In certain examples, the chip kill functionality can be extended to the non-volatile memory device 102 by organizing the data stored in the non-volatile memory device 102 as the data is organized in the volatile memory devices 105 such that each page of the non-volatile memory device 102 includes small, non-continuous chunks of data from only one data lane of the volatile memory devices 105.

[0014] Upon power being re-established to the memory module 100, the non-volatile memory controller 103 can retrieve data from the pages of the non-volatile memory device 102 and route the data to proper volatile memory devices 105. If a page failure occurs, the corrupted data from that page can be transferred from the page to the volatile memory devices 105. As is shown below, if the data is organized within the non-volatile memory device 102 according to the present subject matter, the corrupted data will map to a single data lane of the volatile memory. When the host detects corrupted data in the volatile memory after a transfer from the non-volatile memory, the host can employ the chip kill algorithm and the data can be corrected.

[0015] FIG. 2 illustrates generally an example data lane to non-volatile memory page mapping 200 for a memory module according to various examples of the present subject matter. A non-volatile memory controller can manage data storage such that each page of each die of the non-volatile memory device receives data from only one data lane (Data lane 0...7) of the volatile memory. A data lane of volatile memory can be used to store the ECC information as data is written to or retrieved from the volatile memory devices. In some examples, the register can organize data movement with the host to make sure an ECC data lane (Data lane ECC) is used to store ECC information.

[0016] The data lanes of the volatile memory devices can be organized so that any page (Page 0...21) of any die of the non-volatile memory device includes only data from one data lane of the volatile memory. Such organization, upon failure of a non-volatile memory die, can allow chip kill functionality because even with a page failure of the non-volatile circuit, the chance that the number

of errors within a consecutive chunk of data exceeds the chip kill algorithm recovery capability is reduced considerably. With reference to FIG. 2, an example method to achieve this organization of data is to read and load eight data lanes of data from the volatile memory devices to the non-volatile memory device. For the illustrated example, the eight data lanes can program 64 pages of an example non-volatile memory device that can accommodate 8k bytes of data per page. After programming the 64 pages, the ECC data lane can be read and loaded to another 8 pages of the non-volatile memory device. If a page of data is corrupted because of a non-volatile memory failure, the chip kill functionality, executed by the system host, can allow normal operation of the system after restoring the data to the volatile memory. In addition to providing an example storage scheme that supports chip kill functionality, the example methods of transferring data between the volatile and non-volatile memories can also be more power efficient. Conventional NVDIMM data transfer schemes which can require power expenditures to both blocks of volatile memory. Certain examples of the present subject matter can limit power to a single block during transfer because data is transferred by data lane between the volatile and non-volatile memories of the memory module. In certain examples, addition signaling of the register or the non-volatile memory controller can be used to regulate power from the memory module power supplies such that power distribution can be regulated at the data lane level during data transfers between the volatile and non-volatile memories thereby providing additional power conservation. Additional signaling could, for example, be in the form of clock enable signals or chip enable signals extending from, for example, the non-volatile memory controller to each volatile memory device.

[0017] FIG. 3 illustrates generally an alternative example memory module 300. In certain examples the memory module 300 can include memory devices referred to a non-volatile load-reduced dual in-line memory modules (NVL RDIMM). In certain examples, the memory module 300 can include a power supply 301, a non-volatile memory device 302, a non-volatile memory controller 303, a register 304, a plurality of volatile memory circuits 305 such as dynamic random access memory (DRAM) circuits, a plurality of data buffers 307, and a plurality of interconnects (APR, BPR, LDQA, LDQB, LCOM, BCOM). In certain examples, when a system power supply (not shown) is removed from the memory module 300, the memory device power supply 301 can provide sufficient power to allow data to be moved from the volatile memory circuits 305 to the non-volatile memory device 302. In certain examples, the power supply 301 or backup power source for the memory module 300 can include a battery. In some examples, the power supply 301 can include a capacitor. In certain examples, the memory module is representative of a DDR4 memory module.

[0018] The non-volatile memory device 302 can save

and maintain data when a main power source is removed. Examples of non-volatile memory devices can include, among others, flash memory including NAND or NOR flash memories. In certain examples, the volatile memory circuit 305 can include dynamic random access memory (DRAM) that can provide exceptional read and write speeds for use by a host processor or host system. Unfortunately, in many examples, such volatile memory circuits 305 cannot maintain stored data when system power is removed from the memory module 300 for even a relatively short interval. The register 304 can control access to the volatile memory circuits 305. When system power is present, the register can receive memory requests from the host system via a host command and address bus 308. The register 304 can respond to command and address control information by controlling the volatile memory circuits 305 and the data buffers 307 to present data to the host or receive data from the host for storage in the volatile memory circuits 305. Error correction code (ECC) information from the host can be stored in the volatile memory circuits 305.

[0019] In certain examples, the memory module 300 can include two volatile memory command and address buses (APR, BPR), each bus coupled to a respective block 309, 310 of multiple volatile memory circuits 305. In some examples, the a first block 309 of volatile memory circuits 305 can include five data lanes and a second block 310 of volatile memory circuits 305 can include four data lanes. In certain examples, the register 304 can provide command and address control information via the busses (APR, BPR) to the volatile memory circuits 305 to present and receive data with the host system. In the present example, a data lane is defined as the data communication path associated with transferring data to and from a pair of volatile memory circuits (DRAM X, DRAM X+9) via a corresponding data buffer (DB X).

[0020] In certain examples, the non-volatile memory controller 303 can control movement of data between the volatile memory circuits 305 and the non-volatile memory device 302. For example, upon system power loss, the register 304 can receive control and address information from the non-volatile memory controller 303 via a second control and address bus (LCOM) coupled directly between the register 304 and the non-volatile memory controller 303. Data can be moved from the volatile memory circuits 305 to the non-volatile memory controller 303 using a first data bus (LDQA) and a second data bus (LDQB). The first data bus (LDQA) can be coupled between the first block 309 of volatile memory circuits 305 and the non-volatile memory controller 303 and the second data bus (LDQB) can be coupled between the second block 310 of volatile memory circuits 305 and the non-volatile memory controller 303. In certain examples, the register 304 can provide routing information via a data buffer bus (BCOM) to the plurality of data buffers 307 so that the data is routed to either the first or second data bus (LDQA, LDQB) and not the host system or processor.

[0021] In certain examples, the register 304 and the non-volatile memory controller 303 cooperate to organize the data storage in at least the non-volatile memory device 302 to allow for chip kill functionality to identify chip failures of the non-volatile memory device. Chip kill functionality allows maintaining or reconstructing data of a failed memory chip such as a failed volatile memory chip such that the memory module 300 can continue to function as efficiently as possible with one or more failed volatile memory devices 305. In certain examples, chip kill functionality can be available and maintained by preemptively organizing the data stored in the volatile and non-volatile memory such that each page of non-volatile memory includes data from only one data lane of the volatile memory.

[0022] Upon power being re-established to the memory module 300, the non-volatile memory controller 303 can retrieve data from the pages of the non-volatile memory device 302 and route the data to proper volatile memory devices 305. If a page failure of the non-volatile memory device 302 has occurred, the corrupted data from that page can be transferred from the page to the corresponding volatile memory device 305. As is shown below, if the data is organized according to the present subject matter, the corrupted data will map to a single data lane of the volatile memory. When the host detects corrupted data in the volatile memory after a transfer from the non-volatile memory, the host can employ the chip kill algorithm and the data can be restored.

[0023] FIG. 4 illustrates generally a flowchart of an example method for operating a memory module. At 402, a first plurality of pages of non-volatile memory of the memory module can be filled with data from a first data lane of dynamic random access memory of the memory module. At 404, a second plurality of pages of the non-volatile memory can be filled with data from a second data lane of the dynamic random access memory. The first plurality of pages and the second plurality of pages can be distinct from each other in that the pages do not overlap or share memory.

[0024] At 406, data from an error correction code (ECC) data lane of the volatile memory can be transferred to a third plurality of pages of the non-volatile memory. The third plurality of pages can be distinct from the first plurality of pages and from the second plurality of pages. In certain examples, at 408, the control of the transfer of data from the volatile memory to the non-volatile memory can assure that each page of data of the non-volatile memory includes data from only one data lane of the volatile memory. Such distinction can allow the memory module to have chip kill functionality. Chip kill functionality can allow data to be reconstructed when a memory chip fails such as a volatile memory chip. The present subject matter provides apparatus and methods to extend chip kill functionality to non-volatile memory devices of memory modules such that data can be reconstructed, and restored on failure of a non-volatile memory chip. Upon restoration, the memory module can continue to

operate with the failed chip, thus, preventing unscheduled downtime on the system. For server systems that are counted on to run for extended periods between planned maintenance intervals, the prevention of downtime from unexpected memory failures can provide an overwhelming competitive advantage. In certain examples, filling each page of the non-volatile memory can include transferring data from a first data lane of the volatile memory to the non-volatile memory device via a non-volatile memory controller coupled to the non-volatile memory device. In some examples, filling pages of the non-volatile memory can include isolating a host command and address bus from a register of the memory module. In some examples, the host command and address bus can be isolated from the host using a command and address multiplexer. In certain examples, filling the pages of the non-volatile memory can include coupling the register to a second command and address bus of the memory module using the command and address multiplexer. The second command and address bus can be coupled between the register and the non-volatile memory controller. In some examples, a pair of data buses can be used to transfer data from the volatile memory to the non-volatile memory controller.

[0025] FIG. 5 illustrates generally an example data lane to non-volatile memory page mapping 500 for a memory module according to various examples of the present subject matter. In certain examples, a non-volatile memory controller can manage data storage such that each page of each die of the non-volatile memory device receives data associated with only certain bits of one data lane (Data lane 0...7) of the volatile memory. A data lane of volatile memory can be used to store the ECC information as data is written to or retrieved from the volatile memory devices. In some examples, the register can organize data movement with the host to make sure an ECC data lane (Data lane ECC) is used to store ECC information.

[0026] The data lanes of the volatile memory devices can be organized so that any page (Page 0...21) of any die (DIE 0, ..., DIE 3) of the non-volatile memory device includes data associated with certain bits of one data lane of the volatile memory. In the example of FIG. 5, data associated with the lower four data bits (DQ[0:3]) of a data lane can be programmed to a first page of non-volatile memory and data associated with the upper four data bits (DQ[4:7]) of the data lane can be programmed to a separate second page of non-volatile memory. Such organization, upon failure of a non-volatile memory die can allow chip kill functionality because even with a page failure of the non-volatile circuit, the chance that the number of errors within a consecutive chunk of data exceeds the chip kill recovery algorithm capability is reduced considerably. With reference to FIG. 5, an example method to achieve this organization of data is to read and load eight data lanes of data from the volatile memory devices to the non-volatile memory device. For the illustrated example, the eight data lanes can program 64 pag-

es of an example non-volatile memory device that can accommodate 8k bytes of data per page. The ECC data lane can be read and loaded to another 8 pages of the non-volatile memory device. If a page of data is corrupted because of a non-volatile memory failure, upon power-up of the memory module, the non-volatile memory controller can move data from the non-volatile memory device to the volatile memory devices including attempting to write the data of the failed page to the corresponding data lane. As the host system begins to use the memory module and recognizes corrupted data, the chip kill algorithm of the host can reconstruct the corrupted data and upon restoration of the data, can allow normal operation of the system.

[0027] In certain examples, in addition to providing an example storage scheme that supports chip kill functionality for failed non-volatile memory pages, the example methods of transferring data between the volatile and non-volatile memories can also be more power efficient. Conventional NVDIMM data transfer schemes can require power expenditures to both blocks of volatile memory during the transfer interval. Certain examples of the present subject matter can limit power to a single block during transfer portions of the transfer interval because data is transferred by individual data lane between the volatile and non-volatile memory controller of the memory module. In certain examples, addition signaling of the register or the non-volatile memory controller can be used to regulate power from the memory module power supplies such that power distribution can be regulated at the data lane level during data transfers between the volatile and non-volatile memories thereby providing additional power conservation. Additional signaling could, for example, be in the form of clock enable signals or chip enable signals extending from, for example, the non-volatile memory controller to each volatile memory device.

[0028] FIG. 6 illustrates generally a flowchart of an example method 600 for operating a memory module. At 602, a first page of non-volatile memory of the memory module can be filled with data from the lower bits, such as the lower four bits, of a first data lane of dynamic random access memory of the memory module. This particular example is drawn towards volatile memory devices that can move 8 bits of data at each data lane, however, the scope of the present subject matter is not so limited. At 604, a second page of the non-volatile memory can be filled with data from the upper four bits of the first data lane of the dynamic random access memory. At 606, a third page of non-volatile memory of the memory module can be filled with data from the lower bits of a second data lane of dynamic random access memory of the memory module. At 608, a fourth page of the non-volatile memory can be filled with data from the upper four bits of the second data lane of the dynamic random access memory. In certain examples, each of the first, second, third and fourth pages of the non-volatile memory device can be distinct from each other in that the pages do not overlap or share memory.

[0029] In certain examples, at 610, data from the lower four bits of an error correction code (ECC) data lane of the volatile memory can be transferred to a fifth page of the non-volatile memory. At 612, data from the upper four bits of an error correction code (ECC) data lane of the volatile memory can be transferred to a sixth page of the non-volatile memory. In certain examples, the fifth and sixth pages of non-volatile memory can be distinct from the first thru fourth pages as well as being distinct from each other. In certain examples, at 614, the control of the transfer of data from the volatile memory to the non-volatile memory can assure that each page of data of the non-volatile memory includes data from only one data lane of the volatile memory. Such distinction can allow the memory module to have chip kill functionality or non-volatile memory failures. In certain examples, filling each page of the non-volatile memory can include transferring data from a first data lane of the volatile memory to the non-volatile memory device via a non-volatile memory controller coupled to the non-volatile memory device. In some examples, filling pages of the non-volatile memory can include isolating a host command and address bus from a register of the memory module. In some examples, the host command and address bus can be isolated from the host using a command and address multiplexer. In certain examples, filling the pages of the non-volatile memory can include coupling the register to a second command and address bus of the memory module using the command and address multiplexer. The second command and address bus can be coupled between the register and the non-volatile memory controller. In some examples, a pair of data buses can be used to transfer data from the volatile memory to the non-volatile memory controller.

[0030] The above Description of Embodiments includes references to the accompanying drawings, which form a part of the detailed description. The drawings show, by way of illustration, specific embodiments in which methods, apparatuses, and systems discussed herein can be practiced. These embodiments are also referred to herein as "examples." Such examples can include elements in addition to those shown or described. However, the present inventors also contemplate examples in which only those elements shown or described are provided. Moreover, the present inventors also contemplate examples using any combination or permutation of those elements shown or described (or one or more aspects thereof), either with respect to a particular example (or one or more aspects thereof), or with respect to other examples (or one or more aspects thereof) shown or described herein.

[0031] The flowchart and block diagrams in the figures illustrate the architecture, functionality, and operation of possible implementations of systems, methods and computer program products according to various aspects of the present disclosure. In this regard, each block in the flowchart or block diagrams can represent a module, segment, or portion of code, which comprises one or more

executable instructions for implementing the specified logical function(s). It should also be noted that, in some alternative implementations, the functions noted in the block can occur out of the order noted in the figures. For example, two blocks shown in succession can, in fact, be executed substantially concurrently, or the blocks can sometimes be executed in the reverse order, depending upon the functionality involved. It will also be noted that each block of the block diagrams and/or flowchart illustration, and combinations of blocks in the block diagrams and/or flowchart illustration, can be implemented by special purpose hardware-based systems that perform the specified functions or acts, or combinations of special purpose hardware and computer instructions.

[0032] The functions or processes described herein can be implemented in software, hardware, human implemented procedures, or a combination thereof. The software can consist of computer executable instructions stored on computer readable media such as memory or other type of storage devices. The term "computer readable media" is also used to represent any means by which the computer readable instructions can be received by the computer, such as by different forms of wired or wireless transmissions. Further, such functions correspond to modules, which are software, hardware, firmware or any combination thereof. Multiple functions can be performed in one or more modules as desired, and the embodiments described are merely examples. The software can be executed on a digital signal processor, ASIC, microprocessor, or other type of processor operating on a computer system, such as a personal computer, server or other computer system. In this document, the terms "a" or "an" are used, as is common in patent documents, to include one or more than one, independent of any other instances or usages of "at least one" or "one or more." In this document, the term "or" is used to refer to a non-exclusive or, such that "A or B" includes "A but not B," "B but not A," and "A and B," unless otherwise indicated. In this document, the terms "including" and "in which" are used as the plain-English equivalents of the respective terms "comprising" and "wherein." Also, in the following claims, the terms "including" and "comprising" are open-ended, that is, a system, device, article, composition, formulation, or process that includes elements in addition to those listed after such a term in a claim are still deemed to fall within the scope of that claim. Moreover, in the following claims, the terms "first," "second," and "third," etc. are used merely as labels, and are not intended to impose numerical requirements on their objects.

[0033] The above description is intended to be illustrative, and not restrictive. For example, the above-described examples (or one or more aspects thereof) can be used in combination with each other. Other embodiments can be used, such as by one of ordinary skill in the art upon reviewing the above description. Also, in the above Description of Embodiments, various features can be grouped together to streamline the disclosure. This should not be interpreted as intending that an unclaimed

disclosed feature is essential to any claim. Rather, inventive subject matter can lie in less than all features of a particular disclosed embodiment. Each claim may be considered as a separate embodiment, which may be combined with each other in various combinations.

[0034] The scope of the various embodiments of the invention should be determined with reference to the appended claims.

Claims

1. A method to preserve chip kill functionality comprising:

transferring data stored in volatile memory to non-volatile memory (102, 302), wherein the volatile memory comprises multiple volatile memory devices (105) arranged in multiple data lanes, the respective data lanes comprising a respective data path and respective associated volatile memory circuits (305); wherein each data lane extends across multiple volatile memory devices (105); and wherein the multiple data lanes comprise multiple data lanes in which data is written in the memory circuits (305), and a ECC data lane in which ECC information is written in the memory circuits (305); writing selected lower data bits from a first data lane associated with a first volatile memory device (105) to a first page of the non-volatile memory (102, 302); writing selected upper data bits from a second data lane associated with a second volatile memory device (105) to a second page of the non-volatile memory (102, 302); and writing ECC information from the ECC data lane to a third page of the non-volatile memory (102, 302); and wherein the data of a failed volatile memory device (105) in an individual data lane, as stored in a respective page of the non-volatile memory (102, 302) can be reconstructed through use of the ECC information stored in the third page of the non-volatile memory (102, 302).

2. The method of claim 1, further comprising receiving host command and address information from a host (112, 312) so that each of the first volatile memory device (105) and the second volatile memory device (105) perform normal operations in response to the host command and address information, and wherein each of the writing selected lower data bits from a first data line to a first page and the writing selected upper data bits from a second data lane to a second page is performed in response to a command and address information provided from a memory con-

troller (103, 303) that is distinct from the host (112, 312).

3. The method of claim 2, further comprising selecting one of the host command and address information and the command and address information provided from the memory controller (103, 303) by a command and address multiplexer (106).
4. The method of claim 1, wherein each of the selected lower data bits from a first data lane and the selected upper data bits from a second data lane is transferred to the non-volatile memory (102, 302) through a common data bus.
5. The method of claim 1, wherein the selected lower data bits from a first data lane are transferred to the non-volatile memory (102, 302) through at least a first data bus, the selected upper data bits from a second data lane are transferred to the non-volatile memory (102, 302) through second data bus, and wherein the first data bus and the second data bus are distinct.
6. A system (100, 300) preserving chip kill functionality, the system comprising:

a non-volatile memory including a plurality of memory pages;
 a plurality of volatile memory devices (105) arranged in plurality of data lanes, the respective data lane comprising a respective data path and respective associated volatile memory circuits (305);
 wherein each data lane extends across multiple volatile memory devices (105); and
 wherein the multiple data lanes comprise multiple data lanes in which data is written in the memory circuits (305), and a ECC data lane in which ECC information is written in the memory circuits (305); and
 a controller (103, 303) configured to store data to the plurality of memory pages of the non-volatile memory (102, 302), including:

writing selected lower data bits from a first data lane associated with a first volatile memory device (105) to a first page of the non-volatile memory (102, 302),
 writing selected upper data bits from a second data lane associated with a second volatile memory device (105) to a second page of the non-volatile memory (102, 302), and
 writing ECC information from the ECC data lane to a third page of the non-volatile memory (102, 302); and

wherein the data of a failed volatile memory de-

vice (105) in an individual data lane, as stored in a respective page of the non-volatile memory (102, 302) can be reconstructed through use of the ECC information stored in the third page of the non-volatile memory (102, 302).

7. The system (100, 300) of claim 6, further comprising:

a plurality of first external connectors configured to receive first control information from a host (112, 312); and
 a register (104, 304), and

wherein the controller (103, 303) is configured to provide second control information, and the register (104, 304) is configured to control operations of the volatile memory devices (105) in response to selected one of the first control information and the second control information.

8. The system (100, 300) of claim 6, wherein each of the data lanes includes a data multiplexer (107) coupled to a respective volatile memory device (105).

9. The system (100, 300) of claim 6, wherein each of the data lanes includes a data buffer (307) coupled to a respective volatile memory device (305).

10. The system (100, 300) of claim 6, further comprising a single printed-circuit board, and wherein the non-volatile memory (102, 302), the volatile memory devices (105) and the controller (103, 303) are mounted on the single printed-circuit board.

11. The system (100, 300) of claim 10, further comprising a backup power source (101, 301) mounted on the single printed-circuit board and configured to provide a power to the non-volatile memory (102, 302), the volatile memory device (105, 305) and the controller (103, 303).

12. The system (100, 300) of claim 6, further comprising: a processor configured to perform an error correction operation on data.

Patentansprüche

1. Verfahren zur Wahrung einer Chip-Kill-Funktion, Folgendes umfassend:

Übertragen von in einem flüchtigen Speicher gespeicherten Daten in einen nichtflüchtigen Speicher (102, 302), wobei der flüchtige Speicher mehrere flüchtige Speichervorrichtungen (105) umfasst, welche in mehreren Datenspuren angeordnet sind, wobei die jeweiligen Datenspuren einen jeweiligen Datenpfad und jeweilige zu-

- geordnete flüchtige Speicherschaltungen (305) umfassen;
wobei jede Datenspur sich über mehrere flüchtige Speichervorrichtungen (105) erstreckt; und
wobei die mehreren Datenspuren mehrere Datenspuren umfassen, in welchen Daten in die Speicherschaltungen (305) geschrieben werden, und eine ECC-Datenspur, in welcher ECC-Informationen in die Speicherschaltungen (305) geschrieben werden;
Schreiben ausgewählter unterer Datenbits aus einer ersten Datenspur, welche einer ersten flüchtigen Speichervorrichtung (105) zugeordnet ist, in eine erste Seite des nichtflüchtigen Speichers (102, 302);
Schreiben ausgewählter oberer Datenbits aus einer zweiten Datenspur, welche einer zweiten flüchtigen Speichervorrichtung (105) zugeordnet ist, in eine zweite Seite des nichtflüchtigen Speichers (102, 302);
Schreiben von ECC-Informationen aus der ECC-Datenspur in eine dritte Seite des nichtflüchtigen Speichers (102, 302); und
wobei die Daten einer ausgefallenen flüchtigen Speichervorrichtung (105) in einer individuellen Datenspur, welche in einer jeweiligen Seite des nichtflüchtigen Speichers (102, 302) gespeichert sind, unter Verwendung der in der dritten Seite des nichtflüchtigen Speichers (102, 302) gespeicherten ECC-Informationen wiederhergestellt werden können.
2. Verfahren nach Anspruch 1, ferner umfassend Empfangen von Host-Befehl- und Adresseninformationen von einem Host (112, 312) dergestalt, dass jede der ersten flüchtigen Speichervorrichtung (105) und der zweiten flüchtigen Speichervorrichtung (105) normale Operationen in Reaktion auf die Host-Befehl- und Adresseninformationen durchführen, und wobei jedes des Schreibens von ausgewählten unteren Datenbits aus einer ersten Datenspur in eine erste Seite und des Schreibens von ausgewählten oberen Datenbits aus einer zweiten Datenspur in eine zweite Seite in Reaktion auf Befehl- und Adresseninformationen durchgeführt wird, welche von einer Speichersteuerung (103, 303) bereitgestellt werden, welche sich von dem Host (112, 312) unterscheidet.
3. Verfahren nach Anspruch 2, ferner umfassend Auswählen eines von den Host-Befehl- und Adresseninformationen und den Befehl- und Adresseninformationen, welche von der Speichersteuerung (103, 303) bereitgestellt werden, durch einen Befehls- und Adressen-Multiplexer (106).
4. Verfahren nach Anspruch 1, wobei jedes von den ausgewählten unteren Datenbits aus einer ersten
- Datenspur und den ausgewählten oberen Datenbits aus einer zweiten Datenspur in den nichtflüchtigen Speicher (102, 302) durch einen gemeinsamen Datenbus übertragen wird.
5. Verfahren nach Anspruch 1, wobei die ausgewählten unteren Datenbits aus einer ersten Datenspur an den nichtflüchtigen Speicher (102, 302) durch mindestens einen ersten Datenbus übertragen werden, wobei die ausgewählten oberen Datenbits aus einer zweiten Datenspur in den nichtflüchtigen Speicher (102, 302) durch einen zweiten Datenbus übertragen werden, und wobei der erste Datenbus und der zweite Datenbus unterschiedlich sind.
6. System (100, 300) zur Wahrung einer Chip-Kill-Funktion, wobei das System Folgendes umfasst:
- einen nichtflüchtigen Speicher, welcher eine Vielzahl von Speicherseiten einschließt;
eine Vielzahl von flüchtigen Speichervorrichtungen (105), welche in einer Vielzahl von Datenspuren angeordnet ist, wobei die jeweilige Datenspur einen jeweiligen Datenpfad und jeweilige zugeordnete flüchtige Speicherschaltungen (305) umfasst;
wobei jede Datenspur sich über mehrere flüchtige Speichervorrichtungen (105) erstreckt; und
wobei die mehreren Datenspuren mehrere Datenspuren umfassen, in welchen Daten in die Speicherschaltungen (305) geschrieben werden, und eine ECC-Datenspur, in welcher ECC-Informationen in die Speicherschaltungen (305) geschrieben werden; und
eine Steuerung (103, 303), welche konfiguriert ist, um Daten in der Vielzahl von Speicherseiten des nichtflüchtigen Speichers (102, 302) zu speichern, Folgendes einschließend:
- Schreiben ausgewählter unterer Datenbits aus einer ersten Datenspur, welche einer ersten flüchtigen Speichervorrichtung (105) zugeordnet ist, in eine erste Seite des nichtflüchtigen Speichers (102, 302),
Schreiben ausgewählter oberer Datenbits aus einer zweiten Datenspur, welche einer zweiten flüchtigen Speichervorrichtung (105) zugeordnet ist, in eine zweite Seite des nichtflüchtigen Speichers (102, 302), und
Schreiben von ECC-Informationen aus der ECC-Datenspur in eine dritte Seite des nichtflüchtigen Speichers (102, 302); und
wobei die Daten einer ausgefallenen flüchtigen Speichervorrichtung (105) in einer individuellen Datenspur, wie sie in einer jeweiligen Seite des nichtflüchtigen Speichers (102, 302) gespeichert sind, unter

- Verwendung der in der dritten Seite des nichtflüchtigen Speichers (102, 302) gespeicherten ECC-Informationen wiederhergestellt werden können.
- 5
7. System (100, 300) nach Anspruch 6, ferner Folgendes umfassend:
- eine Vielzahl von ersten externen Verbindern, welche konfiguriert sind, um erste Steuerungsinformationen von einem Host (112, 312) zu empfangen; und
- 10 ein Register (104, 304), und wobei die Steuerung (103, 303) konfiguriert ist, um zweite Steuerungsinformationen bereitzustellen, und das Register (104, 304) konfiguriert ist, um Operationen der flüchtigen Speichervorrichtungen (105) in Reaktion auf eine ausgewählte von den ersten Steuerungsinformationen und den zweiten Steuerungsinformationen zu steuern.
- 15
8. System (100, 300) nach Anspruch 6, wobei jede der Datenspuren einen Daten-Multiplexer (107) einschließt, welcher mit einer jeweiligen flüchtigen Speichervorrichtung (105) gekoppelt ist.
- 25
9. System (100, 300) nach Anspruch 6, wobei jede der Datenspuren einen Daten-Puffer (307) einschließt, welcher mit einer jeweiligen flüchtigen Speichervorrichtung (305) gekoppelt ist.
- 30
10. System (100, 300) nach Anspruch 6, ferner umfassend eine einzelne Leiterplatte, und wobei der nichtflüchtige Speicher (102, 302), die flüchtigen Speichervorrichtungen (105) und die Steuerung (103, 303) auf der einzelnen Leiterplatte montiert sind.
- 35
11. System (100, 300) nach Anspruch 10, ferner umfassend eine Ersatz-Stromquelle (101, 301), welche an der einzelnen Leiterplatte montiert und konfiguriert ist, um Strom an den nichtflüchtigen Speicher (102, 302), die flüchtige Speichervorrichtung (105, 305) und die Steuerung (103, 303) abzugeben.
- 40
12. System (100, 300) nach Anspruch 6, ferner Folgendes umfassend:
- einen Prozessor, welcher konfiguriert ist, um eine Fehlerkorrekturoperation an Daten vorzunehmen.
- 45
- 50
- Revendications**
1. Procédé pour préserver la fonctionnalité de destruction des puces, comprenant :
- 55 le transfert de données stockées dans une mémoire volatile vers une mémoire non volatile
- (102, 302), dans lequel la mémoire volatile comprend plusieurs dispositifs de mémoire volatile (105) disposés dans plusieurs lignes de données, les lignes de données respectives comprenant un chemin de données respectif et des circuits de mémoire volatile (305) associés respectifs ;
- dans lequel chaque ligne de données s'étend sur plusieurs dispositifs de mémoire volatile (105) ; et
- dans lequel les plusieurs lignes de données comprennent plusieurs lignes de données dans lesquelles des données sont écrites dans les circuits de mémoire (305), et une ligne de données ECC dans laquelle des informations ECC sont écrites dans les circuits de mémoire (305) ; l'écriture de bits de données inférieurs sélectionnés à partir d'une première ligne de données associée à un premier dispositif de mémoire volatile (105) sur une première page de la mémoire non volatile (102, 302) ;
- l'écriture de bits de données supérieurs sélectionnés à partir d'une seconde ligne de données associée à un second dispositif de mémoire volatile (105) sur une deuxième page de la mémoire non volatile (102, 302) ; et
- l'écriture d'informations ECC à partir de la ligne de données ECC sur une troisième page de la mémoire non volatile (102, 302) ; et
- dans lequel les données d'un dispositif de mémoire volatile (105) défectueux dans une ligne de données individuelle, telles que stockées dans une page respective de la mémoire non volatile (102, 302) peuvent être reconstruites par l'utilisation des informations ECC stockées dans la troisième page de la mémoire non volatile (102, 302).
2. Procédé selon la revendication 1, comprenant en outre la réception d'informations de commande et d'adresse d'hôte depuis un hôte (112, 312) de sorte que chacun du premier dispositif de mémoire volatile (105) et du second dispositif de mémoire volatile (105) exécute des opérations normales en réponse aux informations de commande et d'adresse d'hôte, et dans lequel chacune de l'écriture de bits de données inférieurs sélectionnés à partir d'une première ligne de données vers une première page et de l'écriture de bits de données supérieurs sélectionnés à partir d'une seconde ligne de données vers une deuxième page est exécutée en réponse à des informations de commande et d'adresse fournies par un contrôleur de mémoire (103, 303) qui est distinct de l'hôte (112, 312).
3. Procédé selon la revendication 2, comprenant en outre la sélection de l'une des informations de commande et d'adresse d'hôte et des informations de

commande et d'adresse fournies par le contrôleur de mémoire (103, 303) par un multiplexeur de commande et d'adresse (106).

4. Procédé selon la revendication 1, dans lequel chacun des bits de données inférieurs sélectionnés d'une première ligne de données et des bits de données supérieurs sélectionnés d'une seconde ligne de données est transféré à la mémoire non volatile (102, 302) par un bus de données commun.

5. Procédé selon la revendication 1, dans lequel les bits de données inférieurs sélectionnés d'une première ligne de données sont transférés à la mémoire non volatile (102, 302) par l'intermédiaire d'au moins un premier bus de données, les bits de données supérieurs sélectionnés d'une seconde ligne de données sont transférés à la mémoire non volatile (102, 302) par l'intermédiaire d'un second bus de données, et dans lequel le premier bus de données et le second bus de données sont distincts.

6. Système (100, 300) préservant la fonctionnalité de destruction des puces, le système comprenant :

une mémoire non volatile comprenant une pluralité de pages de mémoire ;

une pluralité de dispositifs de mémoire volatile (105) disposés dans une pluralité de lignes de données, la ligne de données respective comprenant un chemin de données respectif et des circuits de mémoire volatile (305) associés respectifs ;

dans lequel chaque ligne de données s'étend sur plusieurs dispositifs de mémoire volatile (105) ; et

dans lequel les plusieurs lignes de données comprennent plusieurs lignes de données dans lesquelles des données sont écrites dans les circuits de mémoire (305), et une ligne de données ECC dans laquelle des informations ECC sont écrites dans les circuits de mémoire (305) ; et

un contrôleur (103, 303) configuré pour stocker des données dans la pluralité de pages de mémoire de la mémoire non volatile (102, 302), comprenant :

l'écriture de bits de données inférieurs sélectionnés à partir d'une première ligne de données associée à un premier dispositif de mémoire volatile (105) sur une première page de la mémoire non volatile (102, 302), l'écriture de bits de données supérieurs sélectionnés à partir d'une seconde ligne de données associée à un second dispositif de mémoire volatile (105) sur une deuxième page de la mémoire non volatile (102, 302),

et

l'écriture d'informations ECC à partir de la ligne de données ECC sur une troisième page de la mémoire non volatile (102, 302) ;

et

dans lequel les données d'un dispositif de mémoire volatile (105) défaillant dans une ligne de données individuelle, telles que stockées dans une page respective de la mémoire non volatile (102, 302) peuvent être reconstruites par l'utilisation des informations ECC stockées dans la troisième page de la mémoire non volatile (102, 302).

7. Système (100, 300) selon la revendication 6, comprenant en outre :

une pluralité de premiers connecteurs externes configurés pour recevoir des premières informations de commande d'un hôte (112, 312) ; et un registre (104, 304), et

dans lequel le contrôleur (103, 303) est configuré pour fournir des deuxièmes informations de commande, et le registre (104, 304) est configuré pour commander des opérations des dispositifs de mémoire volatile (105) en réponse à l'une sélectionnée des premières informations de commande et des deuxièmes informations de commande.

8. Système (100, 300) selon la revendication 6, dans lequel chacune des lignes de données comprend un multiplexeur de données (107) couplé à un dispositif de mémoire volatile (105) respectif.

9. Système (100, 300) selon la revendication 6, dans lequel chacune des lignes de données comprend un tampon de données (307) couplé à un dispositif de mémoire volatile (305) respectif.

10. Système (100, 300) selon la revendication 6, comprenant en outre une carte de circuit imprimé unique, et dans lequel la mémoire non volatile (102, 302), les dispositifs de mémoire volatile (105) et le contrôleur (103, 303) sont montés sur la carte de circuit imprimé unique.

11. Système (100, 300) selon la revendication 10, comprenant en outre une source d'alimentation de secours (101, 301) montée sur la carte de circuit imprimé unique et configurée pour fournir une alimentation à la mémoire non volatile (102, 302), au dispositif de mémoire volatile (105, 305) et au contrôleur (103, 303).

12. Système (100, 300) selon la revendication 6, comprenant en outre : un processeur configuré pour effectuer une opéra-

tion de correction d'erreur sur des données.

5

10

15

20

25

30

35

40

45

50

55

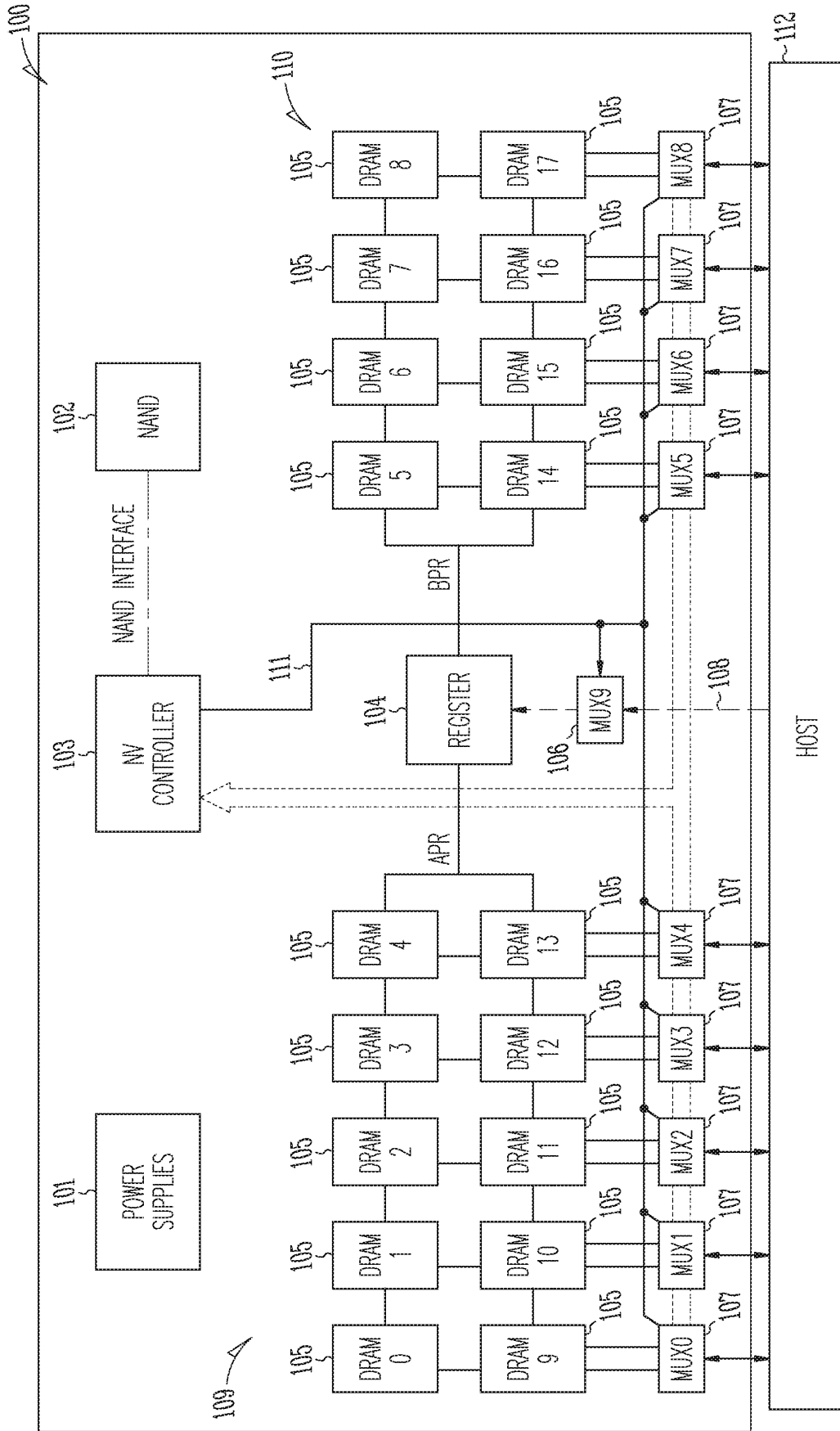


Fig. 1

200

	PAGE 0	PAGE 2	PAGE 4	PAGE 6	PAGE 8	PAGE 10	PAGE 12	PAGE 14	PAGE 16	PAGE 18	PAGE 20
DIE 0	DATA LANE 0 8K BYTES	DATA LANE 0 8K BYTES	DATA LANE 0 8K BYTES	DATA LANE 0 8K BYTES	DATA LANE 0 8K BYTES	DATA LANE 0 8K BYTES	DATA LANE 0 8K BYTES	DATA LANE 0 8K BYTES	DATA LANE ECC 8K BYTES	DATA LANE 0 8K BYTES	DATA LANE 0 8K BYTES
DIE 1	DATA LANE 1 8K BYTES	DATA LANE 1 8K BYTES	DATA LANE 1 8K BYTES	DATA LANE 1 8K BYTES	DATA LANE 1 8K BYTES	DATA LANE 1 8K BYTES	DATA LANE 1 8K BYTES	DATA LANE 1 8K BYTES	DATA LANE ECC 8K BYTES	DATA LANE 1 8K BYTES	DATA LANE 1 8K BYTES
DIE 2	DATA LANE 2 8K BYTES	DATA LANE 2 8K BYTES	DATA LANE 2 8K BYTES	DATA LANE 2 8K BYTES	DATA LANE 2 8K BYTES	DATA LANE 2 8K BYTES	DATA LANE 2 8K BYTES	DATA LANE 2 8K BYTES	DATA LANE ECC 8K BYTES	DATA LANE 2 8K BYTES	DATA LANE 2 8K BYTES
DIE 3	DATA LANE 3 8K BYTES	DATA LANE 3 8K BYTES	DATA LANE 3 8K BYTES	DATA LANE 3 8K BYTES	DATA LANE 3 8K BYTES	DATA LANE 3 8K BYTES	DATA LANE 3 8K BYTES	DATA LANE 3 8K BYTES	DATA LANE ECC 8K BYTES	DATA LANE 3 8K BYTES	DATA LANE 3 8K BYTES
	PAGE 1	PAGE 3	PAGE 5	PAGE 7	PAGE 9	PAGE 11	PAGE 13	PAGE 15	PAGE 17	PAGE 19	PAGE 21
DIE 0	DATA LANE 4 8K BYTES	DATA LANE 4 8K BYTES	DATA LANE 4 8K BYTES	DATA LANE 4 8K BYTES	DATA LANE 4 8K BYTES	DATA LANE 4 8K BYTES	DATA LANE 4 8K BYTES	DATA LANE 4 8K BYTES	DATA LANE ECC 8K BYTES	DATA LANE 4 8K BYTES	DATA LANE 4 8K BYTES
DIE 1	DATA LANE 5 8K BYTES	DATA LANE 5 8K BYTES	DATA LANE 5 8K BYTES	DATA LANE 5 8K BYTES	DATA LANE 5 8K BYTES	DATA LANE 5 8K BYTES	DATA LANE 5 8K BYTES	DATA LANE 5 8K BYTES	DATA LANE ECC 8K BYTES	DATA LANE 5 8K BYTES	DATA LANE 5 8K BYTES
DIE 2	DATA LANE 6 8K BYTES	DATA LANE 6 8K BYTES	DATA LANE 6 8K BYTES	DATA LANE 6 8K BYTES	DATA LANE 6 8K BYTES	DATA LANE 6 8K BYTES	DATA LANE 6 8K BYTES	DATA LANE 6 8K BYTES	DATA LANE ECC 8K BYTES	DATA LANE 6 8K BYTES	DATA LANE 6 8K BYTES
DIE 3	DATA LANE 7 8K BYTES	DATA LANE 7 8K BYTES	DATA LANE 7 8K BYTES	DATA LANE 7 8K BYTES	DATA LANE 7 8K BYTES	DATA LANE 7 8K BYTES	DATA LANE 7 8K BYTES	DATA LANE 7 8K BYTES	DATA LANE ECC 8K BYTES	DATA LANE 7 8K BYTES	DATA LANE 7 8K BYTES

Fig. 2

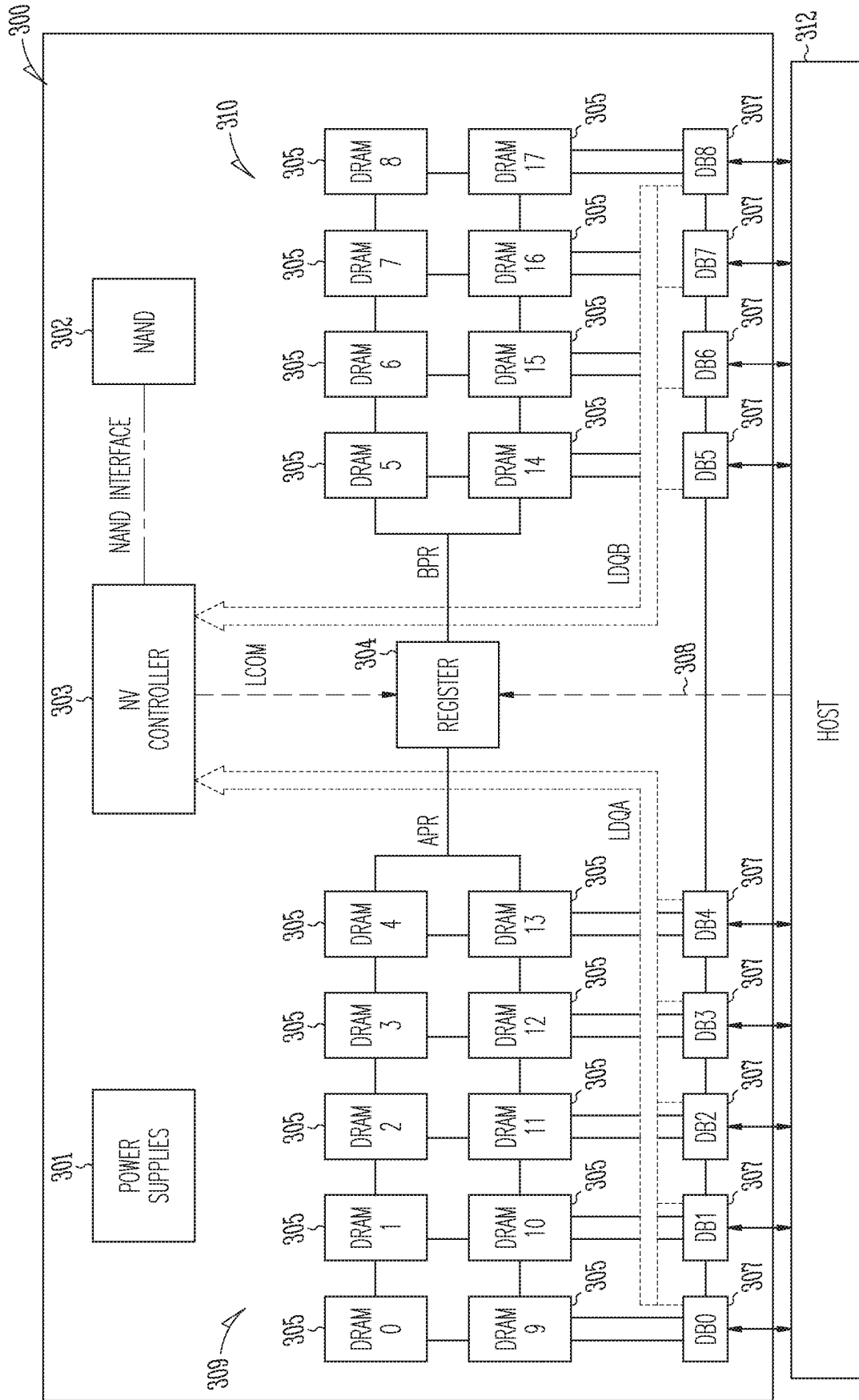


Fig. 3

400 ↗

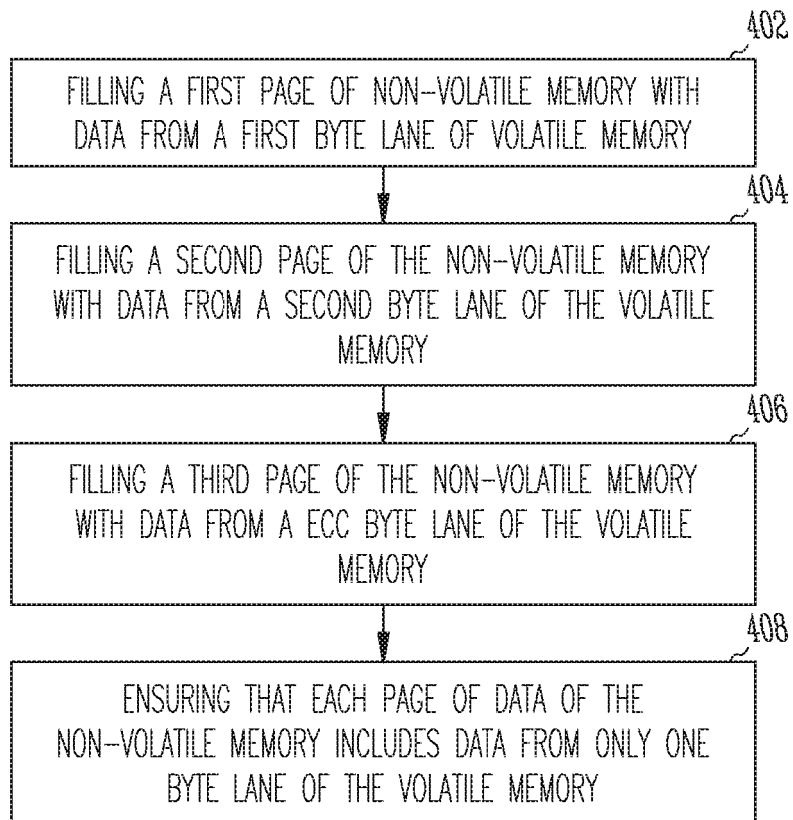


Fig. 4

600 ↗

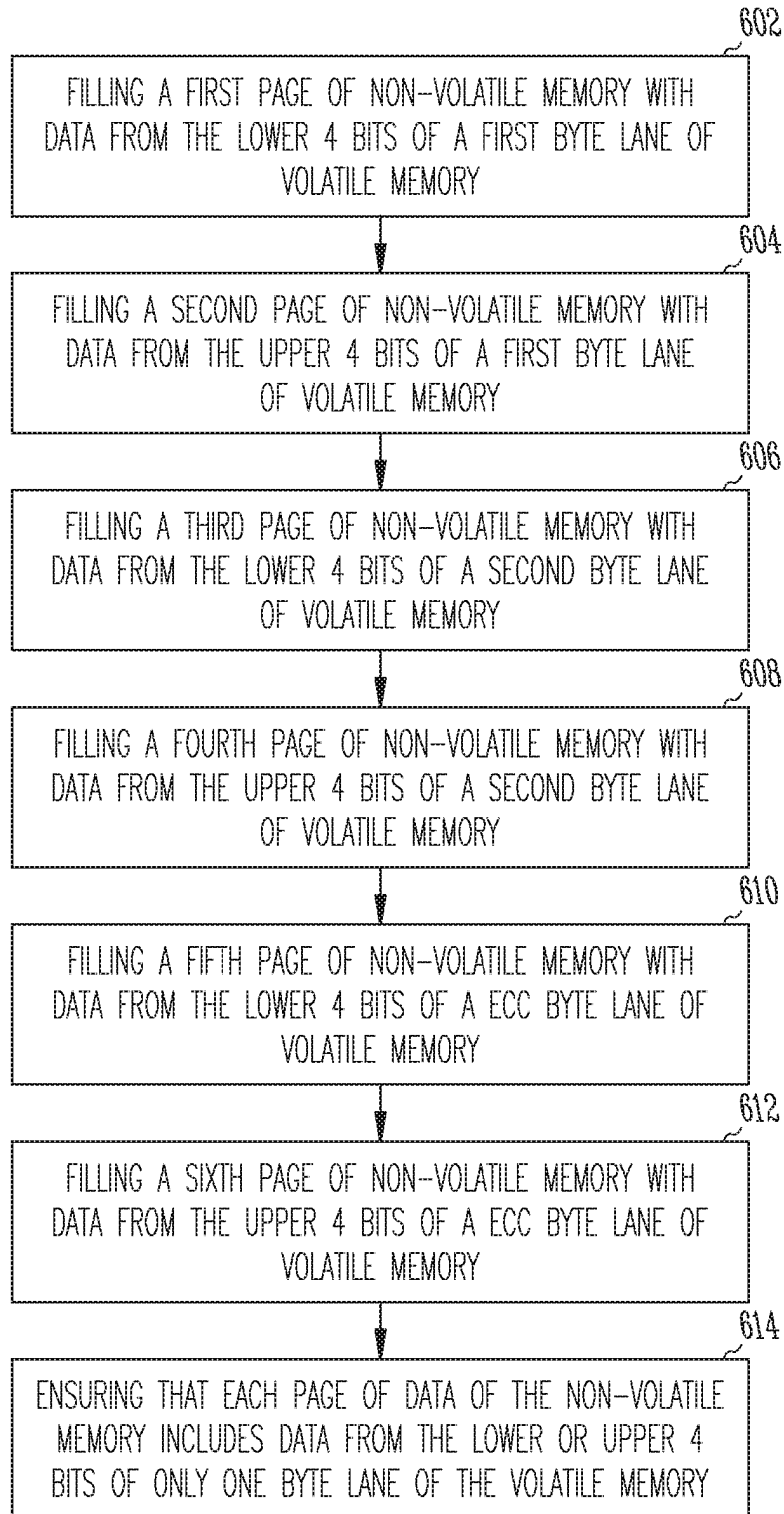


Fig. 6

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- US 00081216 [0001]
- US 20050146939 A [0002]